DOCUMENT RESUME

ED 049 614 24 EM 008 861

AUTHOR	Block, Karen K.
TITLE	Strategies in Computer-Assisted Instruction: A Selective Overview.
INSTITUTION	Pittsburgh Univ., Pa. Learning Research and Development Center.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C. Bureau of Research.
BUREAU NO	BR-5-0253
PUB DATE	70
CONTRACT	OEC-4-10-158(010)
NOTE	47p.
EDRS PRICE	EDRS Price MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Assisted Instruction, Individualized Instruction, Instructional Innovation, Instructional Technology, Interaction, *Intermode Differences, *Programed Tutoring, Substitution Drills, *Teaching Methods
IDENTIFIERS	Learning Research and Development Center

ABSTRACT

The history of some computer-assisted instruction (CAI) strategies is traced. A number of components of computerized instruction systems are described and explanations provided on the influence these components have in the development and production of a CAI system. A description of the interaction between a student and a CAI system is presented to show the impact of CAI on a student. Using the work of Dr. Patrick Suppes at Stanford University and that of the Learning Research and Development Center as primary examples, the instructional strategies of drill-and-practice systems are differentiated from those of tutorial systems. Other modes of CAI, such as simulations and interactive laboratories, are briefly described. Aspects of instructional strategies are considered which bear on the design of CAI lessons. The future of CAI is projected, with special reference to technical problems and curriculum design. (JY)

BR-5-0253
PA-24
EM

UNIVERSITY OF PITTSBURGH - LEARNING R & D CENTER

1

1971/1

STRATEGIES IN COMPUTER-ASSISTED INSTRUCTION:
A SELECTIVE OVERVIEW
KAREN K. BLOCK

STRATEGIES IN COMPUTER-ASSISTED INSTRUCTION:
A SELECTIVE OVERVIEW

Karen K. Block

Learning Research and Development Center
University of Pittsburgh

1970

The research reported herein was supported by the National Science Foundation and by the Learning Research and Development Center supported in part as a research and development center by funds from the United States Office of Education, Department of Health, Education, and Welfare. The opinions expressed in this publication do not necessarily reflect the position or policy of the National Science Foundation or of the Office of Education and no official endorsement should be inferred.

TABLE OF CONTENTS

Strategies in Computer-Assisted Instruction: A Selective Overview

Karen K. Block[1]

Learning Research and Development Center, University of Pittsburgh

I. Computer Applications in Education; Definition of CAI

Current applications of computers in education have led to two major roles which the computer plays in educational systems. The first role is that of management in which the computer serves primarily as a data storage bank and as a data retriever. Computer management systems can serve both at the administrative and at the classroom level. For example, large school systems can store and retrieve data on teacher payrolls, attendance, pupil records, achievement test results, etc.; all of this data handling becomes a less formidable task with the aid of a computer. On the classroom level, the objective of the computer management system is to store and process information on each child and to print out this information in summarized form upon demand, so that it is available for teacher decision making. The computer management system might also contain suggested instructional resources from which the student or teacher might choose the next instructional step. With current systems of computer management of administration or instruction (sometimes called computer-managed instruction--or CMI), the primary function of the computer is the management (storage and access) of large amounts of data.

The second major role the computer plays in education is to serve some function which influences the instructional process itself. Computer

[1] Portions of this monograph will appear in the Encyclopedia of Library and Information Sciences, Volume IV, edited by Alan Kent (New York: Marcel Dekker, Incorporated, in press).

influence in the instructional process comes about through the computer's role as a decision maker as it affects the direction and specification of a student's learning experience. In directing and providing instruction, the computer makes decisions concerning the assessment of a student's current level of mastery of the curriculum, a sequence of paths through the curriculum, and, on a more molecular level, which stimulus should be experienced next within an instructional session. In this latter case, the computer is responsible for the actual instruction experienced by the child in some subject area. In general, then, when a sequence of educational experiences is under computer control, or when the computer plays a role integral to the determination of educational experiences, the words computer assistance characterize these conditions. The contrast between management and assistance is not easily made, since their functions are not easily separated.[2] For example, the choice of data fed to the computer in a management system has implications for the kinds of instructional decisions that are made. However, for our purposes herein, computer management of education refers to data storage and retrieval functions, while computer assistance in education refers primarily to influences on the direction and nature of learning experiences.

With these definitions, computer-assisted education is a rubric for several activities, two of which are computer-assisted testing (CAT or sequential testing), and computer-assisted instruction (CAI). In CAT, the computer chooses test items of varying difficulty as a function of the student's recent successes or failures on related items in order to determine the skills he has mastered in the curriculum. In CAI, the computer is used as a means of instruction, and it determines the chain and nature of learning tasks relevant to some concept or algorithm, etc., of a subject

[2] This point has been made by Cooley and Glaser (1).

matter. The computer can also assist education in additional ways, as in its roles in simulation and in its use as a laboratory tool for courses on varied subject matters. These functions will be elaborated in a later section; at present, we turn our attention to that mode of computer assistance most frequently used: computer-assisted instruction.

In CAI, the computer plays the role of instructor through generating and sequencing learning experiences relevant to some subject matter. In addition, CAI lessons can be and frequently are designed to allow the systematic collection of data relevant to some research hypothesis about optimal conditions for learning. Thus, in addition to providing instruction and producing learning, the computer serves as a research apparatus in which, for example, the effect of certain psychological variables on the amount of learning can be assessed. This monograph will be concerned primarily with CAI as designed for instructional purposes, since there is a paucity of data from research in learning via CAI. At several points, however, some recently published research studies and some on-going research will be cited.

II. CAI in Operation: The System and Student/System Interaction

In order to provide the reader with some notion of CAI as it operates, it is necessary to describe a number of components of computerized instructional systems, and to comment on the functions they perform and the manner in which they influence the development and production of CAI. In addition, a description of a student/system interaction follows so the reader can develop some notion for how a lesson appears to the student.

At the present time, CAI has been implemented on two different types of time-sharing systems: general purpose time-sharing systems and special purpose (for CAI) time-sharing systems. With this, it is not possible to describe a "typical" CAI system configuration in terms of hardware and software components. Also, the large differences in student terminal devices which are presently in use result in stimulus displays with varying characteristics and consoles with a variety of response modes. In turn, the student terminal and the system characteristics jointly influence the nature of the student/system interaction and the lesson characteristics. In this monograph, the dynamics involved in producing CAI are described as follows: the general functional components of a time-sharing system are described, then some specific steps in implementing a CAI lesson on the system are noted, and then one example of a student/system interaction is described.

In general, the hardware of a CAI system consists of a central processing unit (a computer) and a configuration of input/output devices which serve storage functions (magnetic tape, disk) and which provide instruction via their services to the student station devices (the teletype and cathode ray tubes). The software consists of three components; one component is the applications software, programs which transcribe the user requests (i.e., lesson requests from the curriculum designers) into a form that can be used by the computer. A second component is a problem-oriented language which facilitates the conversion of problem-oriented requests of the applications programmer into machine-oriented requests. The third software component is systems software; these are programs which allocate the computer resources into useable services.

To describe CAI implementation, I will explain some procedures and products at the Learning Research and Development Center at the University of Pittsburgh. The Center has been engaged in prototypical CAI work for several years; recently, work began on large scale CAI programs in elementary mathematics and language arts. The CAI staff of lesson designers consists of people with degrees in academic areas such as English, mathematics, and psychology, with some experience and courses in education.

The choice of a topic for a computerize . instructional unit is governed by several factors; of these, the most important involves a judgment of the extent to which computer implementation of the unit provides instructional features optimal for the unit and not possible with traditional instruction. After a topic has been judged well suited for computer implementation, we formulate some notions about how the lessons should proceed at the student terminal. A preliminary discussion with applications programmers (whose task is to transcribe lesson designs to machine executable routines) follows so that any hardware/software constraints on the lesson become apparent. Next, we formalize lesson plans in the form of flow charts; this process requires a careful logical analysis of the events in a lesson and consideration of the priority of decisions and computer options on response outcomes. One example of a lesson plan which has been formalized is an exercise in our spelling program. The student's task is to fill in a letter in a blank space of a spelling word typed by a teletype; the letter to be filled in is in the first letter position on which the student made an error on the first request for a spelling of the word. This CAI spelling exercise features a teletype with an audio component. The child sits in front of the teletype and is instructed by the audio to make some response by pressing the keys on the teletype keyboard. The flow chart in Figure 1 illustrates the sequence of events.

The flow chart is read from top to bottom. After entry into the lesson, the computer types out the word on the teletype with the letter position of the child's first error shown as a blank (show fill-in). Next, a prerecorded audio message is accessed by the computer and is delivered over a speaker or earphones directing the child to "Fill in the missing

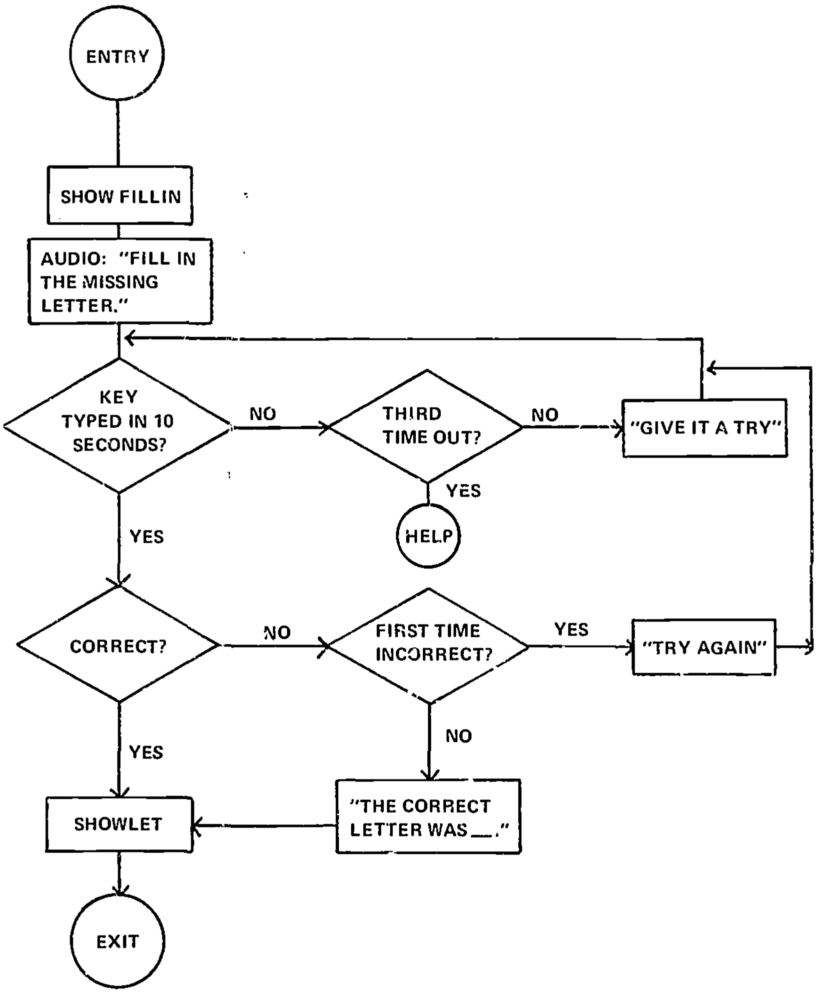


Figure 1: Flow Chart of a Spelling Exercise in a CAI Spelling Program.

letter." If there is no response within 10 seconds, the child is told to "Give it a try." If there is no response within 30 seconds, the program is suspended, and the child is instructed to call the teacher for help (Help). If there is a response within the 30-second period, the computer evaluates the response; if it is correct, the letter is shown in the word (Showlet). If the letter is incorrect, the letter is suppressed, i. e., it is not printed in the word, and the child is instructed to "Try again." If the correct letter is not typed on a second try, the teletype fills in the correct letter along with the audio message, "The correct letter was __."

Lesson implementation continues as the group of applications programmers reviews our flow charts as in Figure 1 with the result being detailed questions about lesson design in an attempt to eliminate logical errors or inconsistencies. After this, the group designs flow charts which are then reviewed and accepted by us. At this stage, coding for the computer begins. (Coding involves the translation of the flow chart into computer executable instructions.) After coding, periods of program testing and modification result in final program acceptance, a point at which the program is run with students.

In order to run the spelling program, the student/system interface employed is a teletype, and the system has audio capabilities which, in our case, consist of a unit with rapid random access to a number of prerecorded audio messages. There are, as noted, other student terminal configurations in use across the country.[3] Most of the present terminal equipment used has been adopted from business communications applications.

One of the first special purpose CAI systems developed in the country was the IBM 1500 system. It has, at a student station, equipment consisting of a cathode ray tube (CRT) display, a typewriter keyboard and light

[3] For a review of student interfaces currently in use, see Hickey (2).

pen, an image projector with a capacity of 1000 frames which may be randomly accessed under computer control, audio equipment (earphones), and microphones. The CRT offers a rapid display of both printed and graphic information to the student; its surface is sensitive to the location of the light pen response (as when the child locates the answer to a multiple-choice question on a display) and also allows a graphic response. An image display and/or audio capability for lessons are often necessary instructional features for some subject matters. Ways in which these features are used in CAI lessons will be described in a later section on a reading program implemented at Stanford.

III. Origin of CAI

CAI is the most recent development in educational technology. Major factors motivating its development have been delineated by Atkinson and Wilson (3) and are reiterated herein. The first factor stems from the capability of the computer for making rapid and sophisticated real time decisions. This capacity means that the computer can make moment-to-moment decisions on the basis of student response characteristics within an instructional session which affect the nature and sequence of that instruction. In essence, the computer can provide individualized instruction of a complexity and sophistication never before possible. Thus, CAI extends the notion of branching that was found within programmed instruction. As an instructional methodology, CAI embodies the general faith among educators that instruction which capitalizes upon and is tailored to individual differences in interests, motives, and learning style is to be preferred over classroom lockstep instruction.

A second factor contributing to the development of CAI was the progress made in computer technology and system development. The introduction of time-sharing systems made CAI economically feasible. Increased hardware reliability and improved man/machine interfaces (e.g., cathode ray tube display devices) contributed to the development of the notion that a computer system might be used for educational purposes. Also, software innovations such as the development of problem-oriented languages allowed for easier implementation of problems and more extensive use of the system, and, taken with the above mentioned factors, contributed to the development of CAI on a large scale basis.

A third factor, as noted by Atkinson and Wilson, is the increased aid to education provided by the funding agencies (e.g., U.S. Office of Education, National Science Foundation) of the Federal Government and private foundations (e.g., the Carnegie Foundation) for purposes of research and development of CAI. In addition, support has been provided for curriculum development by several publishing houses which can provide large scale production and dissemination of CAI lesson materials.

Thus, with the developing technology, the funds for research, development, and implementation, and avenues for dissemination plus a raison d'etre stemming from the work of Skinner and others in the psychology of learning, CAI has become a reality. A statement on the extent of CAI is reported by Atkinson and Wilson (3). They report that in the 1967-1968 school year "several thousand students ranging from elementary school to university received a significant portion of their instruction in at least one subject area under computer control. In Stanford projects alone, approximately 3000 students were processed daily (page 4)." In a later section of this monograph, applications of CAI across the country in elementary and secondary schools and universities will be discussed.

IV. Instructional Strategies in CAI

A lesson or program logic which consists of a collection of rules determining the nature of a subject's interaction with the computer is called an instructional strategy. In particular, instructional strategies control the manner in which the stimulus on trial n+1 is determined by response characteristics on trial n or response characteristics on some past subset of trials. Instructional strategies range in complexity from response insensitive strategies, in which the subject's trial-by-trial path through a lesson can be specified completely in advance of the lesson, to response sensitive strategies, in which characteristics of past responses are taken into account in lesson branching decisions. There are three commonly used response sensitive strategies in CAI: drill-and-practice, tutorial, and dialogue CAI.[4] While all three strategies are response contingent, they differ in terms of the depth of computer involvement in instructional decision making. Levels of computer involvement in decision making can be identified by considering certain features of the decisions and by considering those features in combination as they produce a small number or a large number of logically possible paths through a lesson. One feature to be considered is whether or not decisions occur on a moment-to-moment basis within an instructional session; that is, whether decisions occur on-line as the subject is actively involved in responding to a lesson, or whether branching decisions are made off-line after analyses of responding in a lesson have been made. On-line decisions frequently result in a large variety of paths through a lesson, any one of which has a low probability of occurrence. Another factor determining the complexity of computer involvement in instructional decision making is the characteristics of responding which are used in the decision. Decisions may be dependent upon a single characteristic of the most recent

[4] Following Suppes' (4) classification.

response (e.g., whether or not it is correct), multiple characteristics of the most recent response (its latency, on what trial it first occurred), or, at the most complex level, multiple characteristics of selected subsets of responses. A third factor involves consideration of the complexity of the decision rule itself, after the elements for the decision have been gathered. The rule may involve a simple checking of a cumulative yes/no counter or it may involve a complex mathematical computation, the result of which determines the stimulus on the next trial. These features which determine the extent of computer involvement in instructional decision making are not exclusive, since, e.g., decisions made on a single response characteristic frequently do not involve complex computer manipulations for presentation of the next stimulus item; nevertheless, they do represent the major dimensions along which the three response sensitive instructional strategies are commonly differentiated, and are the major factors that determine the number of logically possible paths.

Another distinction among drill-and-practice, tutorial, and dialogue CAI lies in the extent to which the content and method of the lesson are oriented to teaching basic skills and concepts involved in competent performance required in mastery of subject matter. In order to illustrate the three strategies, several examples from projects in the country are given.

V. Drill-and-Practice CAI

One early large scale drill-and-practice CAI system was initiated in 1964 in elementary schools in California and was directed by Professor Patrick Suppes of Stanford.[5] The drill-and-practice hardware configuration

[5] A book by Suppes, Jerman, and Brian (5) provides a detailed description of all phases of the Stanford drill-and-practice project.

consisted of a large central processor (the DEC PDP-1) with a high-speed drum and teletypes at student stations.

The lessons provide drill and practice in the basic computation skills of arithmetic, and they are organized on the basis of a number of concept blocks at each grade level. The exercises within the concept blocks provide practice on arithmetic skills such as finding sums from 11-60, long division, percent, etc. The student enters a concept block by first taking a pretest which then determines at which of five levels of difficulty he will work. After difficulty level is determined, the student experiences a number of exercises relating to the concept block and some review problems on concept blocks that he previously found difficult. At the end of each day's work, the student's performance is evaluated in an off-line update, and a decision is made concerning the appropriate level of difficulty at which the child should work on the succeeding day. Thus, the child can move among the five levels of difficulty throughout the week. After work on the block is completed, the child is given a posttest which assesses his performance. Performance on that concept block is entered into the student's history file. Review items are selected by the computer from those concept blocks in the student's total past history on which the child did least satisfactorily. Exercises from these blocks are planned for inclusion in his daily lessons on succeeding concept blocks.

In one instructional session, the child is presented with a series of problems from a concept block on a teletype. The problem is typed out with a blank for the response. The child types his answer to a problem and it is evaluated; if he is incorrect, he is permitted to try again; if he is correct, he is presented a new problem. The flow chart in Figure 2 illustrates the item-to-item progression of the Stanford drill-and-practice program.

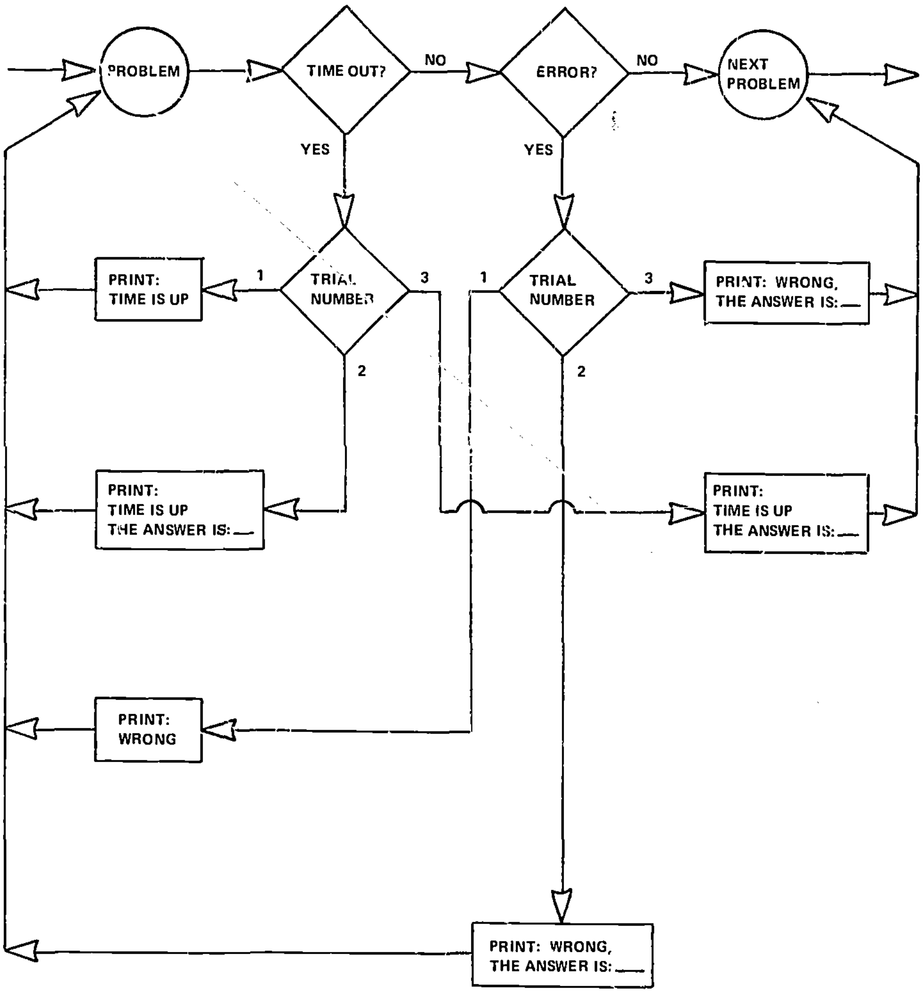

PROBLEM
TIME OUT?
NO
ERROR?
NO
NEXT PROBLEM
YES
YES
PRINT: TIME IS UP
1
TRIAL NUMBER
3
1
TRIAL NUMBER
3
PRINT: WRONG, THE ANSWER IS:
2
2
PRINT: TIME IS UP THE ANSWER IS:
PRINT: TIME IS UP THE ANSWER IS:
PRINT: WRONG
PRINT: WRONG, THE ANSWER IS:

Figure 2: Flow Chart of the Drill-and-Practice Program Logic for Presentation of Problems and Classification of Responses. (From Suppes, Jerman, and Brian [5], p. 27)

As it can be seen, the branching logic is not complex, and the choice of item i+1 is not a function of the student's response on item i on any moment-to-moment basis (or within an instructional session). Also, there is no attempt to tutor the child in the basic mathematical operations. The purpose of the drill-and-practice program is to supplement the regular curriculum taught by the teacher who is responsible for the introduction and development of concepts and ideas; the program provides review and practice. Computer implemented instructional features which provide a kind of individualized instruction in this drill-and-practice mode include a self-paced problem presentation, immediate feedback, and response contingent lesson paths from problem to problem and from one difficulty level to another within a concept block, and from one concept block to another via individualized reviews.

A second example of a drill-and-practice program is a spelling program developed by the Learning Research and Development Center. The program uses an answer processing routine that is more complicated than that in Suppes' drill program, and also uses a response contingent item-spacing routine; these two factors combine with self-pacing and immediate feedback to produce a drill-and-practice program which features greater computer involvement in instructional decision making. The spelling program uses a teletype as the terminal device and runs on the LRDC Executive System which is based on a 32K PDP-7 computer configuration. The audio portion of the program uses a Westinghouse Crow, a unit with rapid access to prerecorded audio messages.

The program can be described rather briefly. The child attempts to learn a list of spelling words. The child is first asked to spell a word, and he attempts to spell it in the context of a meaningful sentence which appears on the teletype. This exercise serves as a pretest for the word; on the basis of the child's response on the pretest, he is branched to one of several paths of instruction which vary according to length and kind of practice. The response characteristics used in response evaluation are number of errors and latency (i. e., time to make a response). If the child makes more than one error or does not respond for a period of 30 seconds, he is branched to the longest series of exercises on the word. If the response contains one error subsequently corrected by the child, or if the word is correctly spelled but not completed within some allotted period of time (a "slow" evaluation--usually five seconds times number of letters in the word), the child experiences a shorter series of exercises, but does receive some practice. If the word is spelled correctly and "fast," then the child receives much less practice on the word. With a correct and fast evaluation, he may be required to perform one exercise before the posttest, or, it is equally likely he may experience no practice at all and proceed directly to the posttest. (The posttest follows every teaching sequence and is similar in form to the pretest.) The latter "pretest-to-posttest" condition was included in the program to test the function of no practice versus some practice on number of trials to a correct spelling. Internal to the program is a provision for spaced review of each word. If a word has been spelled incorrectly on the pretest, it is presented for review after three other words (either new or review) have been presented. In other words, three other words intervene between the $\underline{n}$th and ($\underline{n}$+1)st presentation of a word. This scheme is maintained as much as possible in the program. When a word has been spelled correctly once, on a posttest, it is dropped from the list of words to be reviewed. The flow chart in Figure 3 illustrates the branching logic of the LRDC spelling program.

As it can be seen, the length and kind of practice vary as a function of the response, Path 1 providing the most practice and Path 4 providing

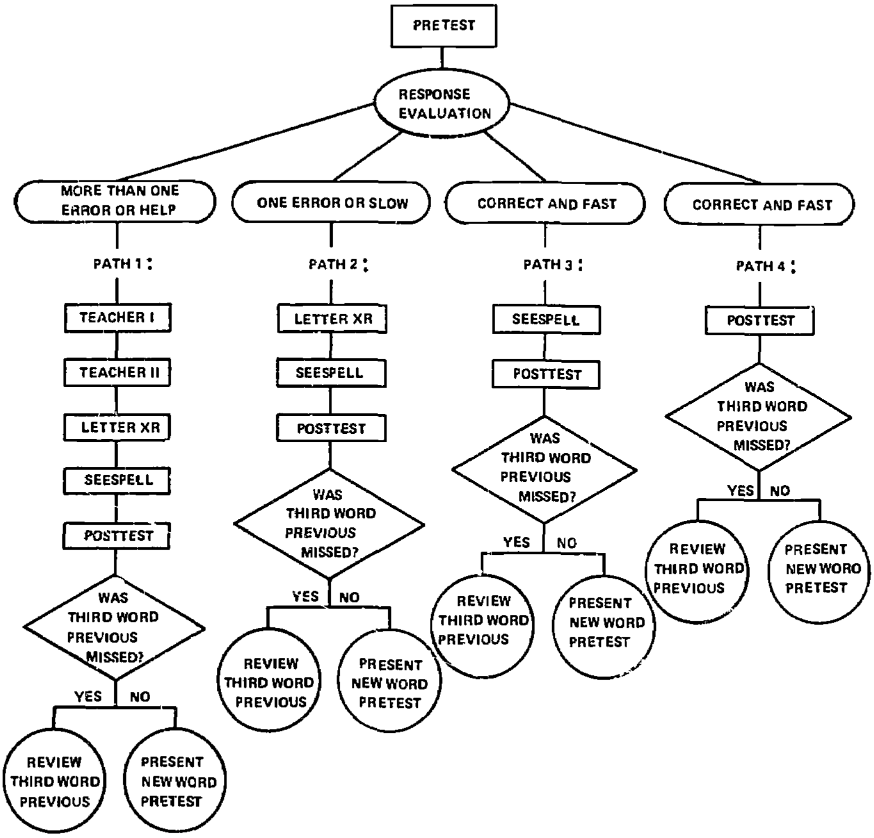


Figure 3: The Branching Logic of the LRDC Spelling Program. Rectangular Boxes Are Different Exercises for the Student to Perform.

the least.[6] The Teacher I exercise spells the word letter by letter and requires the student to copy-type the word; Teacher II requires the child to type the word in the absence of visual prompts. The Letter exercise requires the child to fill a letter in the word; the word is typed for him with a blank at the letter position of his first error in the pretest or an arbitrary position if the response was slow or if no error was made. The Seespell is a discrimination exercise; the child must choose the correct spelling from highly confusible distract words.

Thus, when the nature of the decisions made by the computer in determining an instruction sequence in this spelling program is considered, it is clear that these decisions are sensitive to more than one response characteristic, that these decisions must be made on-line within an instructional session, and that the subject's response on pretest and review trials determines subsequent exercise characteristics and amount of exercise. In terms of the extent of computer involvement in instructional decision making, it is clear that the program provides a deeper level of interaction between the subject and the computer than the Stanford drill-and-practice program. However, the LRDC spelling program does not provide instruction in the basic skills required for spelling competence; e.g., there is no attempt to teach phoneme-grapheme mappings. The proper use of the program is in conjunction with the classroom instruction in spelling rules and generalizations. Thus, the program remains a drill-and-practice program with some complex response monitoring on the part of the computer.

[6] Clearly the program has been designed under the assumption that response quality is a reliable and valid indicator of the length of practice required for learning and subsequent retention of a spelling word. This assumption is currently under experimental scrutiny in our Phase One Spelling program remoted to the Oakleaf School in the Baldwin-Whitehall School District near Pittsburgh, Pennsylvania.

VI. Tutorial CAI

At a second or deeper level of interaction between the computer and the student are tutorial CAI systems. The name "tutorial" is reminiscent of face-to-face learning interaction between a student and a skilled tutor. This, to some extent, is what is simulated in tutorial CAI systems; these systems take responsibility for the introduction and development of basic concepts and skills in subject areas. This simulation of teacher-student relationships is typically accompanied by complex branching schemes which allow more flexibility in lesson content and style and are implemented via lesson logics that are more complex than those found in drill-and-practice CAI. The depth of computer involvement is increased over that in a drill-and-practice program by virtue of: (1) the presence of moment-to-moment instructional decisions made on the basis of the student's response history; (2) answers processed taking account of several characteristics of recent or remote responses (e.g., latency, magnitude, sequence); and (3) the presence of response generated stimuli resulting from mathematical transformations of the student's response, or resulting from a student's choice of stimuli for succeeding trials. With these characteristics, it can be seen that the student's path through a tutorial CAI lesson is, in general, much less predictable a priori than paths through drill-and-practice lessons.

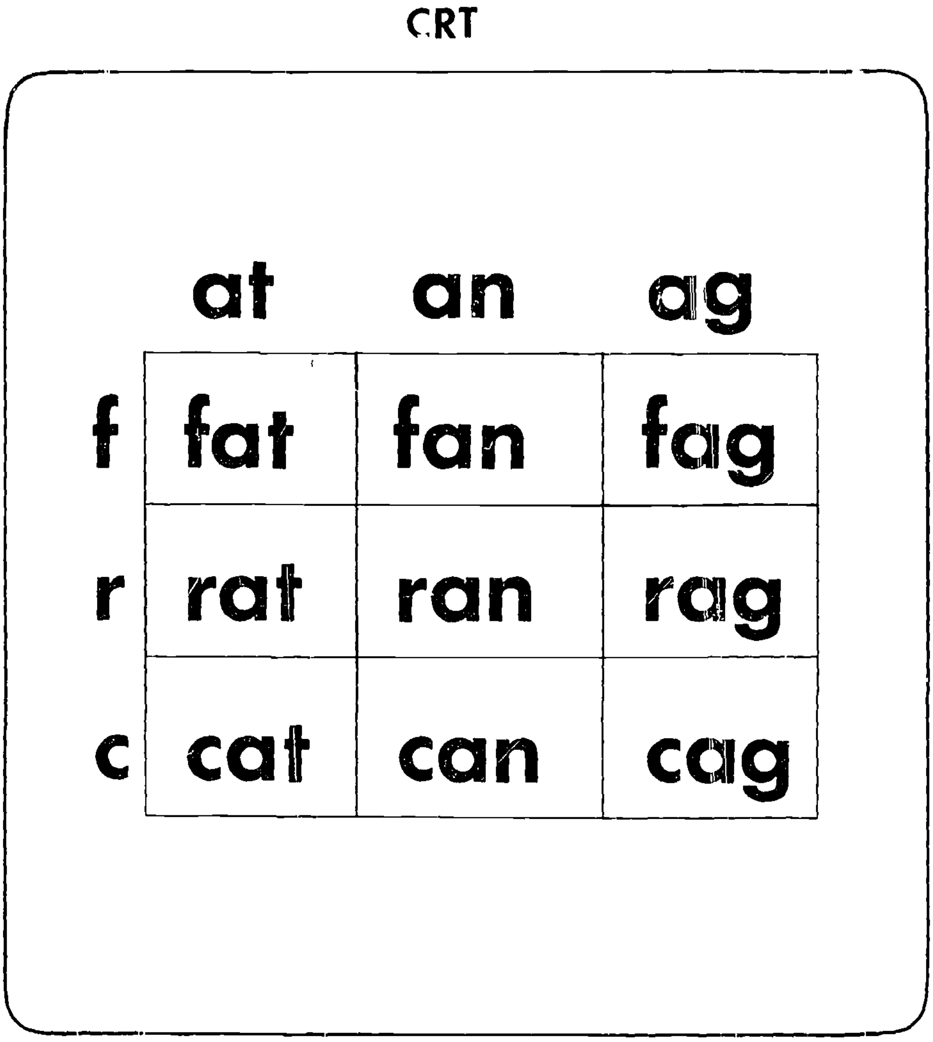
One of the earliest tutorial CAI systems to run in an on-going school environment was the Stanford-Brentwood CAI project in East Palo Alto, California. The project was directed by Dr. Richard Atkinson of the department of psychology at Stanford. The curriculum taught via CAI was reading, with attention to two skills essential for reading capability: decoding and comprehension. Examples from computerized lessons are given in some detail in Atkinson (6) along with an elaboration of the rationale of the curriculum and some early results. Only a brief sketch of the curriculum and system is given here.

The hardware configuration of the Stanford Tutorial System (the IBM 1500 system) was described in an earlier section. It was developed by Stanford and IBM and was more complicated than that of the drill-and-practice system. It provided audio with graphic and printed displays at the student interface.

The CAI reading curriculum developed by the Stanford group is jointly based on an analysis of the logical organization of the subject matter and on the psychology of learning. Atkinson (6) has characterized the approach to the curriculum as one using applied psycholinguistics. One task in the CAI reading curriculum which is especially interesting and appears quite relevant to the teaching of basic skills in reading is called the "matrix construction" task. It is elaborated here to illustrate one type of tutorial CAI.

The matrix construction task is one used to teach decoding skills. In particular, the task provides practice in mapping orthographically similar patterns to their appropriate rhyming or alliterative sound patterns. To illustrate the problem domain, an example of a completed matrix is given in Figure 4. In this case, the columns contain alliterations, and the rows contain differing initial sounds.

Each cell is taught in the same fashion. If the words to be taught are consonant-vowel-consonant letter patterns as in the example, the word is divided into an initial unit (the beginning consonant and a row of the matrix) and a final unit (vowel and ending consonant). The task of the child is to choose a row unit to combine with a column unit to produce a word, or, more generally, to produce a letter pattern containing certain sound features. The child chooses the word to be formed from a list of words by touching the light pen to the surface of the CRT. The choices available to the child diagnose the type of error the child is making. Errors are diagnosed as being due to difficulty in identifying the initial unit (he has



	at	an	ag
f	fat	fan	fag
r	rat	ran	rag
c	cat	can	cag

Figure 4: A Constructed Matrix from the Stanford Reading Program. (From Atkinson [6], p. 230)

chosen a word with an incorrect initial unit but a correct final unit), in identifying the final unit (the word chosen contains an incorrect final unit), or in identifying both.

Succeeding frames which the child experiences are contingent upon the type of errors he has made. To illustrate how it goes, consider construction of the first cell of a matrix. Figure 5 contains the flow chart for forming a cell in the matrix construction task. The Parts A, B, C, and D are defined in Figure 6. Let RR1 stand for response request 1 delivered via audio; CA is the event of a correct answer; and WAi is the event of a wrong answer on the ith try. The screen shows what appears on the visual CRT display.

To explain, in Part A the child touches one of the array of response choices; if he is correct, he proceeds to Part D where he receives one additional practice trial. If, however, the child makes an error, then the nature of the error (on initial unit, final unit, or both) branches him to remedial Parts B or C or both B and C. When a student has made a correct response on Parts A and D, he is advanced to the next cell as in Figure 7.

Cells are added in random order; cell building is continued until the matrix is complete. A posttest onsists of asking the child to locate the cell of the matrix containing a word pronounced by audio. Errors are categorized as before, and remedial work specific to error categories is given if the number of errors exceeds some predetermined criteria for mastery in these categories.

This computerized instruction in reading is tutorial; the lesson is aimed at the development of basic skills, and the lesson content is flexible as a consequence of the error-contingent branching scheme. Branching is also dependent upon two dimensions of the response (initial and final), and branching decisions are made on the basis of subsets of the response history. In addition, the path an individual takes through the series of exercises is not entirely predictable at the start of a lesson.

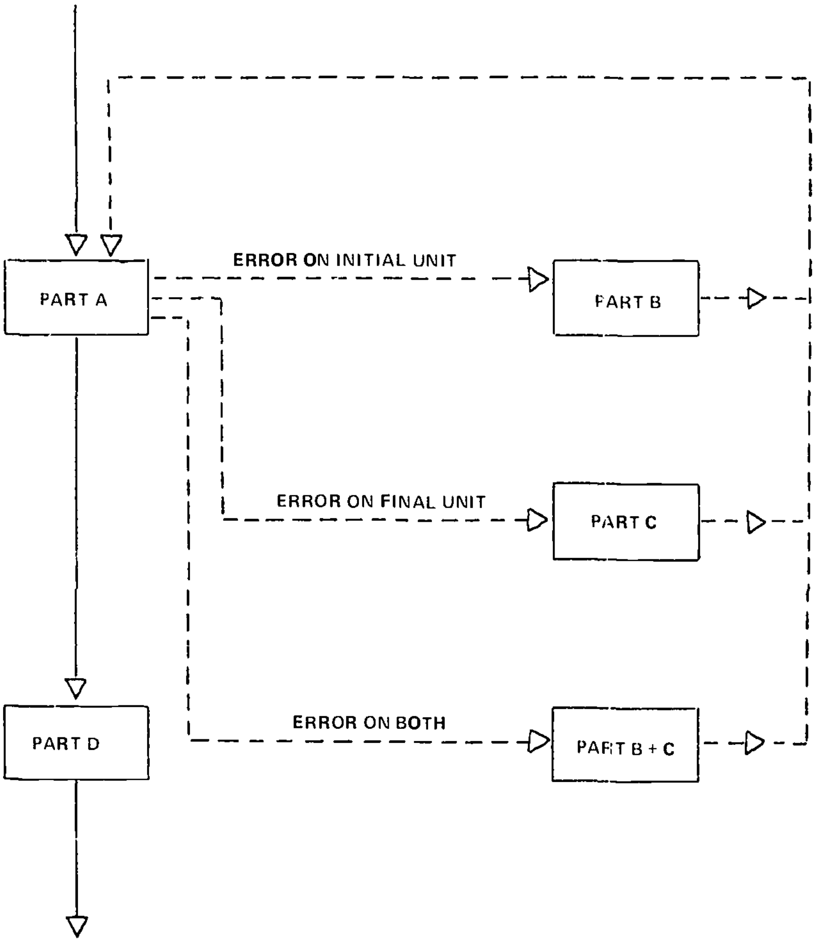
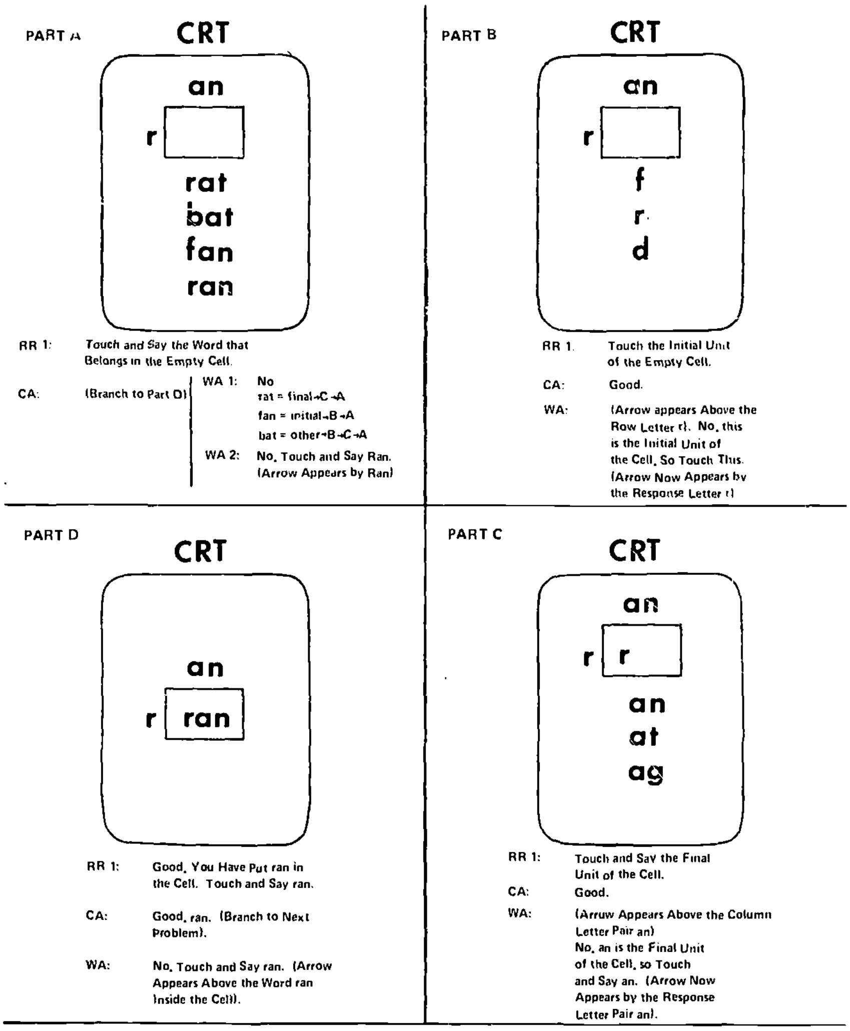


Figure 5: Flow Chart for the Construction of a Cell in the Matrix. (From Atkinson [6], p. 228)



PART A

RR 1: Touch and Say the Word that Belongs in the Empty Cell.

CA: (Branch to Part D)

WA 1: No
rat = final→C→A
fan = initial→B→A
bat = other→B→C→A

WA 2: No. Touch and Say Ran. (Arrow Appears by Ran)

PART B

RR 1. Touch the Initial Unit of the Empty Cell.

CA: Good.

WA: (Arrow appears Above the Row Letter r). No, this is the Initial Unit of the Cell, So Touch This. (Arrow Now Appears by the Response Letter r)

PART C

RR 1: Touch and Say the Final Unit of the Cell.

CA: Good.

WA: (Arruw Appears Above the Column Letter Pair an) No, an is the Final Unit of the Cell, so Touch and Say an. (Arrow Now Appears by the Response Letter Pair an).

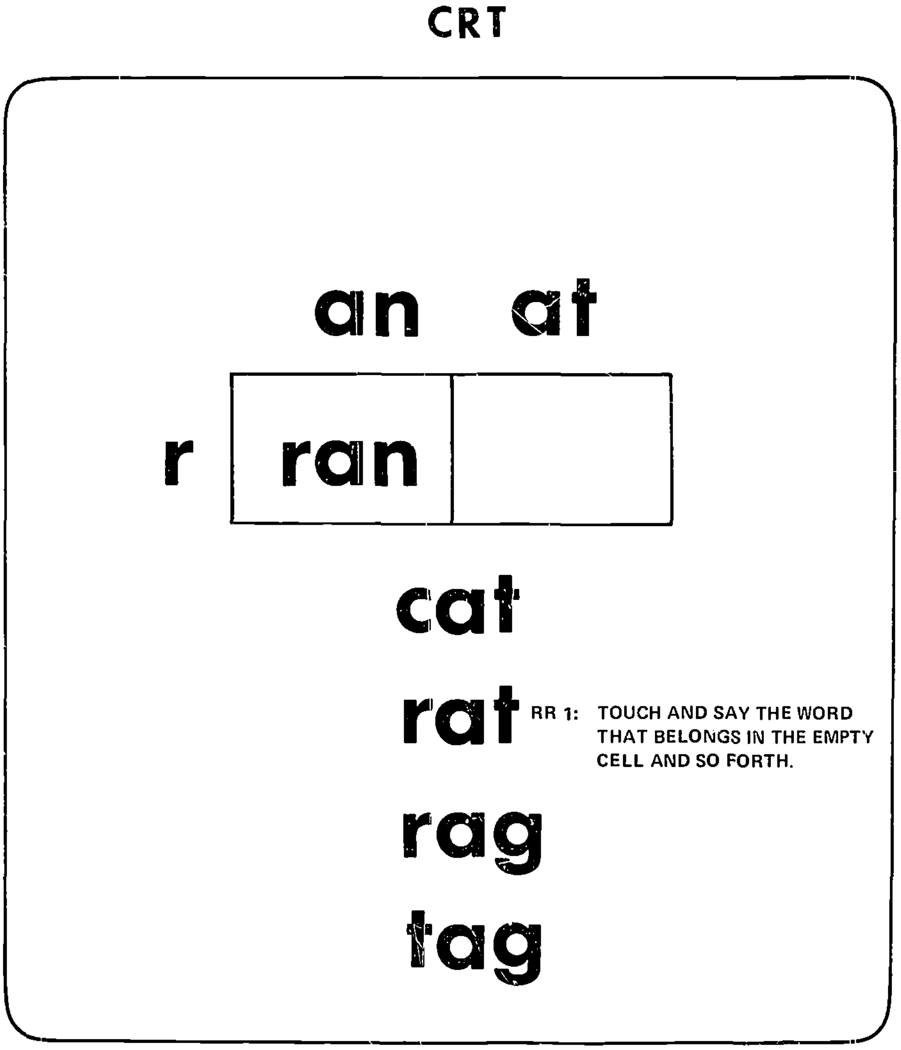
PART D

RR 1: Good, You Have Put ran in the Cell. Touch and Say ran.

CA: Good, ran. (Branch to Next Problem).

WA: No. Touch and Say ran. (Arrow Appears Above the Word ran Inside the Cell).

Figure 6: First Cell of the Matrix Construction Task.
(From Atkinson [6], p. 229)



CRT

	an	at
r	ran	

cat

rat RR 1: TOUCH AND SAY THE WORD THAT BELONGS IN THE EMPTY CELL AND SO FORTH.

rag

tag

Figure 7: Addition of the Next Cell in the Matrix Construction Task. (From Atkinson [6], p. 230)

At the Learning Research and Development Center, we have under development a series of lessons on set theoretic concepts which are tutorial in nature. There is, however, a difference in style between these CAI mathematics lessons and the aforementioned lessons in reading. In particular, in our tutorial CAI in mathematics, the learner has the option to control the instructional sequence; he can choose the task he wishes to perform, and the stimuli which appear as a consequence of his response are mathematical functions of several dimensions of that input response. Our mathematics lessons are not exclusively in a question-and-answer mode; rather, the flow of instruction is frequently controlled by a stimulus-generation subroutine which is initiated by the subject's previous response. Because of the great degree of learner control over the instructional sequence, there are a great number of sequences or paths through the material that are possible; hence, any one path has a low probability of occurrence. With the large number of path options and absence of a strict question-and-answer mode, instruction appears, at least to the learner, to be a "freer" interchange with the computer, more adaptive and less constraining than that found in frame-by-frame CAI.

In order to illustrate this brand of tutorial CAI, a discussion of a lesson in set theory, called ALPHASETS, is given below. The flow charts and sample printouts presented in the discussion include most of the aforementioned characteristics. The ALPHASETS lessons run from the LRDC Executive System; the student sends and receives messages at a teletype.

The objective of the ALPHASETS lesson is to teach a child how to form all possible partitions of sets having a non-prime number of elements such that the partitions consist of equivalent subsets, e.g., a set with eight elements can be divided into two subsets of four elements, four subsets of two elements, eight subsets of one element and the set itself, one subset of eight elements. Also, practice on this concept is provided across a wide sample of set sizes. There are a number of small loops which teach prerequisite concepts and skills (e.g., the concept of a set partition) and provide the child with practice on the machine so he learns the "language" of the lesson--e.g., that the teletype symbol "[" is a ready signal for the child to begin typing a set; that a "?" indicates some response constraint has been violated (e.g., in the ALPHASETS lesson, set elements must be nondistinct alphabetic characters); also, that subsets of a set are formed by using a spacebar response to group the elements of the set, etc. In Figure 8, the flow chart for the generation and completion of a list of set partitions is shown.

Parts A, B, and C in Figure 8 are portions of the subject/computer interaction during the generation of set partitions and are shown in Figures 9, 10, and 11 as sample printouts. The notation used in the flow chart and Parts A, B, and C distinguishes subject input from computer output. In this notation, RRi stands for a response request, a message printed on the teletype which asks for an input from the student; VRi is a valid response (that is, one which satisfies the response constraints of the lesson and is a correct response); IRi is an invalid or incorrect response; and CRi is a computer response to a student input (computer responses result in set partitions or take the form of prompts to aid the student in completing the list of partitions). TO is a time out; the student has not responded within the time allotted for a response. The index i is used to indicate that the occurrence of classes of subject and computer responses is not strictly time based or strictly sequence or content contingent; rather their occurrence is a function of the history of the subject/computer interaction along the time, sequence, and content dimensions.

The exercise in forming a list of all possible set partitions with equivalent subsets begins at Part A. In Part A, in Figure 9, the teletype prints a response request which permits the child to choose between

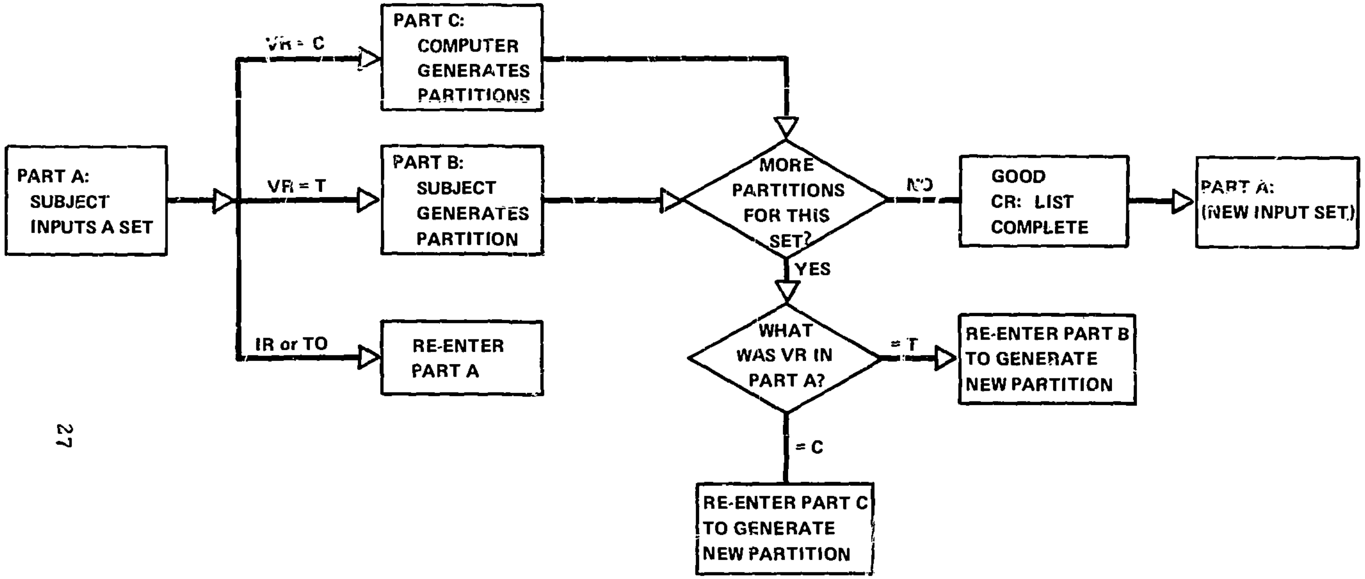
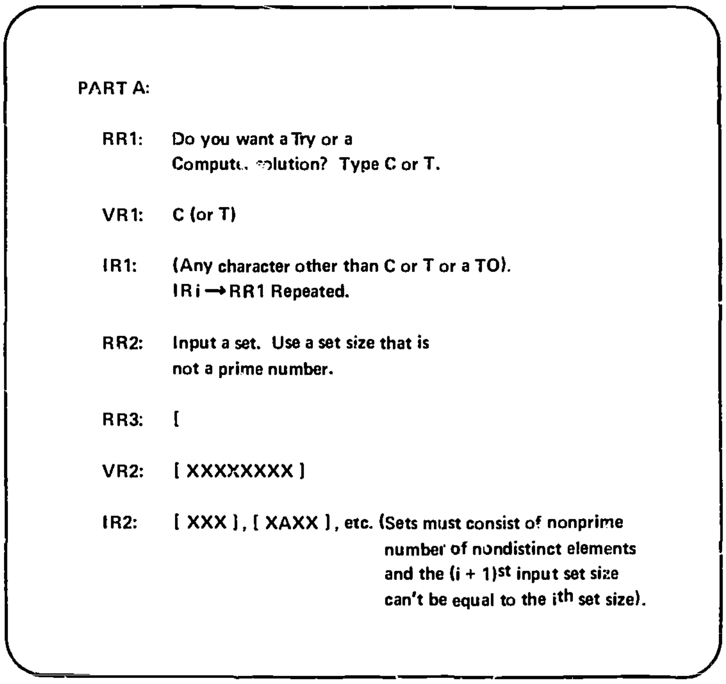

PART A: SUBJECT INPUTS A SET
VR = C
VR = T
IR or TO
PART C: COMPUTER GENERATES PARTITIONS
PART B: SUBJECT GENERATES PARTITION
RE-ENTER PART A
MORE PARTITIONS FOR THIS SET?
NO
YES
GOOD CR: LIST COMPLETE
PART A: (NEW INPUT SET)
WHAT WAS VR IN PART A?
= T
= C
RE-ENTER PART B TO GENERATE NEW PARTITION
RE-ENTER PART C TO GENERATE NEW PARTITION

Figure 8: Flow Chart for the Partitioning of a Set into All Possible Partitions Having Equivalent Subsets. Rectangular Boxes Are Exercises for Which Sample Printouts Are Shown.



PART A:

RR1: Do you want a Try or a
Computer solution? Type C or T.

VR1: C (or T)

IR1: (Any character other than C or T or a TO).
IRi → RR1 Repeated.

RR2: Input a set. Use a set size that is
not a prime number.

RR3: [

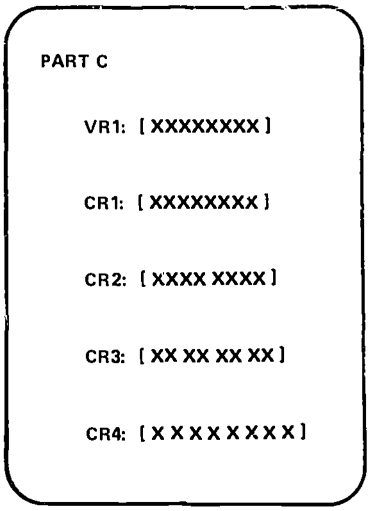
VR2: [XXXXXXXX]

IR2: [XXX], [XAXX], etc. (Sets must consist of nonprime
number of nondistinct elements
and the (i + 1)st input set size
can't be equal to the ith set size).

Figure 9: Sample Subject/Computer Interaction in the Set Input Phase of ALPHASETS Along with Comments on the Program Logic.

generating partitions of a set or computer generation of these. After the child has satisfied the response request, he is asked to input a set upon which he wishes some operations. After the child types in the set, if he has chosen computer generation, the computer generates all possible partitions with equivalent subsets, and they are displayed on the teletype as shown in Part C in Figure 10. On the other hand, if he has chosen to generate the partitions himself, the ready signal appears, and he begins generating partitions. Sample printouts from Part B, shown in Figure 11, are representative cases of the subject/computer interaction which takes place as the subject generates partitions. As indicated, both individual set partitions and the list of partitions are evaluated with respect to allowable part size (e g., a set of size 8 does not contain equivalent subsets of size 3), and, similarly, for number of parts (too many or too few allowable parts appear), and for distinct partitions (the subject may input a partition already present or omit one from the list). Computer responses prompt the subject's response and aid in list completion. These computer responses are a function of the partitions previously generated and act to prompt partitions to be generated. When the list of partitions is completed, the subject is given a "list completed" message and returns to Part A to partition a new set.

This interactive mode of tutorial CAI also characterizes some CAI logic programs authored by Suppes and his staff (7). In brief, in these programs, the student is required to perform logical and algebraic proofs. He has under his control a number of logical and algebraic operators which he can apply to logic and algebraic expressions. After a command has been chosen, the computer carries out the operation if the command has been properly specified and does not violate the rules of logic or algebra. If there is some error in the student's response, the computer types out various error diagnostics.



```
PART C

     VR1:  [ XXXXXXXX ]

     CR1:  [ XXXXXXXX ]

     CR2:  [ XXXX XXXX ]

     CR3:  [ XX XX XX XX ]

     CR4:  [ X X X X X X X X ]
```

(Partitions are formed and printed by the computer)

Figure 10: Sample Computer Printout for Partitioning a Set of Size 8.

PART B:

ALL PARTITIONS CORRECT: VR1: [XXXXXXXX] RR1: [VR1: [XXXX XXXX] (or any of four partitions) RR2: [VR2: [XX XX XX XX] (or any of the remaining three partitions). RR3: [VR3: [XXXXXXXX] (and so forth)	ERROR ON A PARTITION: Part sizes equal and divide set size evenly, but too many or too few parts. VR1: [XXXXXXXX] RRi: [IRi: [XX XX XX XX XX] CRi: Check Number Parts RRi + 1: [IRi + 1: [XX XX XX] CRi + 1: TRY NUMBER PARTS = J (Where number parts = J has not been used in list of partitions)	ERROR ON A PARTITION: Either Part Sizes Unequal or Part Sizes Equal, But Total Number of Elements Incorrect. VR1: [XXXXXXXX] IRi: [XXXXX XXX] or [XXX XXX] CRi: C' eck Part Size IRi + 1: [XXX XXX XX] CRi + 1: Try Part Size = I (where part size = I not yet used in list of partitions)
ERROR BETWEEN PARTITIONS: Most recent partition repeats one previously formed VR1: [XXXXXXXX] RRi: [IRi: [XXXX XXXX] CRi: Last Partition Already in List RRi + 1: [IRi: (any invalid R or TO) CRi + 1: TRY J PARTS WITH I ELEMENTS IN EACH PART (where I, J have not been used in the list of partitions).	TIME OUT BETWEEN PARTITIONS OR WITHIN A PARTITION: VR1: [XXXXXXXX] RRi: [(TIME OUT) CRi: TRY J PARTS WITH I ELEMENTS IN EACH PART. (where I, J have not been used in list of partitions) RRi + 1: [	

Figure 11: Sample Subject/Computer Interactions in ALPHASETS When the Subject Partitions a Set of Size 8.

The types of CAI with the aforementioned generally "freer" interactive style come very close to the mode remaining after drill-and-practice and tutorial CAI have been considered. This mode is called dialogue and is construed to be a free interchange of questions and answers between student and computer: a "conversation." This interchange takes place in natural language; i. e., the student can input a free form question like "Why are whales called mammals?" and can expect to receive some amount of information addressed to the question. Thus, at this third and deepest level of interaction, the purpose is to allow the student to conduct a genuine dialogue with the computer which results in, theoretically, an infinite number of paths to be taken in learning about any one subject. As Suppes (4) points out, dialogue systems, at this time, currently exist at the conceptual rather than the operational level. Since a dialogue between teacher and student is the aim, then some investigators feel that the student should be able to input questions in "natural" response modes; i. e., to use spoken speech or handwritten requests. Thus, the computer system must be able to process spoken speech and handwriting; there are at present a large number of technical difficulties involved in providing this feature. In addition to these difficulties, the problem of analyzing meaning in processing natural language provides another formidable obstacle to development of dialogue systems. Some progress is being made both in hardware development and in processing "restricted" English sentences, but it is doubtful that dialogue systems will be realized in the near future. Furthermore, it is not clear that systems permitting "dialogue" between student and computer will increase the returns of instruction above those possible in the kinds of CAI currently available.

VII. Additional Modes of CAI: Computers as Tools in Instruction

The computer plays varying roles in assisting education; thus, there are different modes of CAI. Thus far, one mode of CAI has been discussed, that in which drill-and-practice and tutorial procedures lead to simulated teacher-student interactions, and the computer functions as the source of instruction. Other modes of CAI are currently in use; these will be discussed somewhat more briefly as they are less frequently employed.

The computer's capability for handling large amounts of data, its capacity for complex problem solving, and the presence of logically defined constraints placed upon its use have been focused toward educationally relevant ends. For one, both SDC (Systems Development Corporation) and LRDC have developed a computer based statistical laboratory in which students can manipulate large data bases to analyze experimental data, or to monitor on-line generation of sampling-distributions.

The use of games and simulations in which the computer implements complex response contingent operations is currently being explored by some projects in the country. Several "simulated environments" have been developed for the purpose of teaching economics. In one project (8), a Sumerian game was developed in which the student is given a characterization of economic conditions in a Sumerian city-state, and is asked to make decisions about the allocation of economic resources. Environments are produced as a function of the student's decisions, and as the student masters each situation, the game increases in complexity with the addition of new factors such as changing social organizations, the addition of technological innovations, etc. The student's objective is to maintain survival of the city-state as factors which influence economic stability vary in increasingly complex fashion.

Another interesting use of the computer as a simulator is that developed by Lagowski at the University of Texas (9). CAI was applied in university chemistry classes in several ways. For one, it was used to demonstrate separation techniques and to illustrate chemical principles in quantitative analysis. The student had a list of commands of analysis actions, e.g., add, wash, etc., corresponding to laboratory operations. The student could apply these as he wished and observe the results of the operations typed or displayed on slides. If the student made errors, he was permitted to repeat the relevant part of the testing scheme. In a second use of CAI, the computer was programmed to simulate a complex piece of equipment (e.g., a spectrometer) which is not ordinarily available to students. In this way, students could become familiar with equipment usually reserved for research purposes. CAI was also used in time compression or expansion experiments. In these, the student could investigate certain kinetic parameters of experiments which proceed too quickly or too slowly for observation purposes through computer simulation.

In the health professions, CAI has been used in several projects across the country in medical diagnosis. In several programs, the instruction in diagnosis occurs in Socratic form (10). Patients' histories and symptoms are given. The student can ask for laboratory tests and additional information of the patient and make preliminary and final diagnoses. Comments on student strategies of problem solving and on student diagnosis are offered by the computer.

Use of the computer, per se, has served as the subject of instruction in some projects in the country. The teaching of computer programming can be useful in teaching mathematics because of the problem-solving orientation required and the necessity for adherence to the rules of logic. Students must analyze their problem, draw flow charts, and write documented programs in computer languages. Some teaching of computer programming to elementary school children has also been used in an attempt to give children insight into key mathematical concepts (e.g., the notion of function and variable) and into heuristics. The language, LOGO,[7] used with these young children is simpler in form than most languages, and bypasses their technical difficulties (e.g., array assignment) to expose connections between mathematics and programming. The children can define new operations so they are permitted to enrich and extend the basic operations permitted, thus encouraging the learning of heuristics in mathematics.

While the preceding several paragraphs do not exhaust the modes of CAI to be found about the country, they give some feeling for the potential variety of roles for the computer in instruction. CAI can be found in a number of locations spanning the United States teaching a large variety of subject matters. Computer-assisted instruction began in the university centers: Stanford University, Florida State University, the University of Texas at Austin, the University of Pittsburgh, the University of Illinois. Some industrial centers have CAI under development (e.g., IBM, RCA). On the university level, such topics as physics, chemistry, mathematics, medical and dental diagnosis, other health sciences, electrical engineering, and programming languages have been taught via CAI. For the same age group, CAI can be found in adult education and vocational training schools.

CAI has also moved into the public schools in New York City; Philadelphia; Palo Alto, California; Pontiac, Michigan; Altoona and Pittsburgh, Pennsylvania; and others. Subjects such as mathematics, logic, reading, and biology have been prepared for the computer, and development will begin soon in other curricular areas. A survey of the literature of applications of CAI can be found in a book by Hickey (2) and major articles concerned with these applications in a book edited by Atkinson and Wilson (12).

[7] Feurzeig et al. discuss the characteristics of LOGO in a final report (11).

VIII. The Design of CAI Lessons

In the previous sections on the use of computers in instruction and the various kinds of instructional strategies, no consideration was given to factors determining the design of a CAI lesson and the choice of an instructional strategy. These factors are relevant to characterizing the current state of CAI curriculum design, and will influence most importantly the future of CAI in education. This section on lesson design treats these considerations.

Designing CAI lessons is a joint function of an analysis of the subject matter to be taught, the objectives to be reached, and psychological considerations about the nature of learning. To begin writing a CAI lesson, the lesson designer must decide upon content: a domain of the curricula for which the lesson is to be designed. An analysis of the subject matter and the skills involved in domain objectives yields some relationships among these objectives in terms of which objectives must be mastered before succeeding objectives can be learned. With at least the content objectives and potential acquisition sequences somewhat well defined, the lesson designer must choose an instructional strategy which decides a sequence of learning tasks or, alternatively, the presentation of stimulus items.[8] Since one aim of CAI is to maximize expected level of achievement of the students, the lesson designer has the task of defining rules of item presentation which provide the greatest instructional return.

[8] What should be made explicit here is that a hierarchy of objectives does not necessarily determine an instructional strategy, i.e., the hierarchy does not necessarily imply the mode in which objectives should be learned for maximal achievement.

Thus, considerations of optimal learning conditions and optimal instruction sequences are of great importance to the lesson designer.[9]

At this time, there are generally two approaches to the design of lessons for optimizing achievement. One is an empirical/intuitive approach in which consideration is given to the learning tasks in terms of their analogues to tasks studied in the psychological laboratory. The tasks are designed by taking into account variables known to speed up or slow down learning in their laboratory analogues, oftentimes with some intuitive concomitant adjustments for motivation effects, the introduction of novelty, provision for transfer to related tasks, etc. To cite a specific example, in LRDC's spelling program, there is provision for "spaced review" of spelling words (i.e., several items intervene between the _n_th presentation of a spelling word and the (_n_+1)st presentation). This feature was incorporated into the program since research in paired-associates learning has demonstrated the superiority of spaced trials for retention (14), and research in CAI spelling has suggested superior retention occurs with larger spaced intervals between item presentations (15).[10]

A second way to design optimal presentation strategies is to adopt a theory or model of learning which has support in the psychological literature and to derive optimal strategies from its axioms. This technique has been used to optimize certain criteria in list learning experiments (see, e.g., Groen and Atkinson [16]), but with modification and extension could be applied to more complicated learning situations, particularly for

[9] Glaser (13) provides an analysis of features relevant to the design of optimal learning sequences in CAI.

[10] The interval of spacing maintained in the spelling program was chosen on the basis of intuitive considerations and on management considerations. It might not be optimal but is probably better for retention than massing item presentations (e.g., repeating a word until it is spelled correctly and not presenting it again).

lesson design in CAI. These two techniques are those that have been most frequently envoked in the CAI lesson design which has some basis in learning research.

At this point, it should also be noted that the programming language used to implement CAI lessons does place large constraints upon lesson design and determines what heretofore has been called CAI style. To some extent then, lesson design can be a joint function of subject matter/psychological considerations and computer programming language constraints. With some programming languages, such as the Coursewriter language implemented on the IBM 1500 system, less effort is required to implement a frame-by-frame kind of CAI, but greater effort is required to produce highly interactive lessons having very sophisticated answer processing.[11]

With respect to the topic of lesson design, it most properly rests on the results of research oriented to learning and instructional problems. Research on variables relevant to CAI continues around the country building a firmer, more extensive base for lesson rationale.

IX. The Future of CAI

In order to project the future of CAI, it is necessary to consider the current state of the art jointly with respect to the state of computer technology and the state of instructional design.

There are a number of hardware problems in CAI which, to date, have not obtained wholly satisfactory solutions. No generally satisfactory operating mode for random-access tape devices has been designed; current

[11] For a discussion of CAI languages and their effect on lesson designers and lesson style, see Zinn (17).

devices are unsatisfactory because search time for prerecorded messages is long enough to be annoying; fidelity of the messages oftentimes is quite poor; and reliability of the unit is a problem. While digitized audio has relieved some of the preceding problems, the expense involved in using it remains a drawback. However, these audio problems may be relieved within several years as a result of work being carried on at various centers.

Another important component of instruction at a CAI student terminal is the visual display. There are a variety of technical problems associated with the production of visual displays at a student terminal. The nature of the problems is different depending on whether the display is produced locally or if it is generated centrally and broadcast to remote terminals. If displays for CRT's are generated centrally, then a special cable with a broad bandwidth is required for information transmission; the result of this is a considerable increase in cost. If the images are generated locally, then currently available devices, such as random-access projectors using Kodak carousels, have a somewhat limited capacity and are expensive when devoted to terminal usage.

In addition to transmission problems, demands for enriched displays with additional display dimensions such as color are increasing. New developments in technology must occur to solve this and the aforementioned major problems associated with visual displays. There are some investigators, e.g., Bitzer at the University of Illinois, who have made significant developmental efforts which promise some solutions in the area of visual displays.

Thus, advances in technology will provide students with the kind of terminal that has features deemed important for quality instruction by instructional designers. Unfortunately, as sophisticated terminal equipment is developed for use, the cost per terminal hour increases considerably. At this date, costs are high even with the simplest systems and are above those of traditionally assisted instruction. However, as standardization of equipment begins and as hardware passes from a developmental to a production stage, it can be expected that these costs will decrease considerably. For a detailed and projected cost analysis of CAI, see Kopstein and Seidel (18).

While problems of cost and technological development are factors relevant to the future of CAI, another factor determining its future is the quality of curriculum design. The future of CAI depends greatly on the extent to which lesson designers can provide the conditions optimal for learning and retention. These conditions can only be provided by a large scale effort in psychological research and theory which attends to variables relevant to instructional design (e.g., sequencing of items, feedback parameters, etc.). At the current time, curriculum design has no specific theory of instruction to guide its development; some investigators have provided frameworks for instructional design or provided procedures in instructional design, but most of recent curriculum design is a product of quasi psychological/intuitive thinking. In order to fulfill the promise of CAI, education needs an instructional theory much like that of stimulus sampling theory in psychology. Stimulus sampling theory offers psychology an approach to theorizing about learning and a structure which can be formalized as models which yield quantitative predictions in many learning situations. Instructional theory needs an analogous development: a trans-situational theory which leads to the formulation of well-defined instructional strategies which provide quantifiable returns.[12]

Thus, when the current state of curriculum design and instructional theory is considered it is clear that it is indeed premature to evaluate the effectiveness of CAI in instruction. However, some recent work in

[12] Somewhat the same case has been stated as in Suppes (19).

psychology and the psychology of instruction by Groen and Atkinson (16), Restle (20), Karush and Dear (21), and Suppes (22) indicates that competent and skillful psychologists have begun attending to the problems of instruction. Consequently, future psychological research into educational processes can be expected to increase because of these early attempts by respected scientists and, also, because of the reward associated with the application of science to social problems. In addition, psychological research and theory in educational processes is a complex and challenging endeavor sure to attract some of the most qualified people in psychology.

There are a number of additional features to CAI which heretofore have not been mentioned, but should be noted as they are relevant to any attempt to judge CAI efforts in education:

1. CAI will make education more scientific. With the amount of control and monitoring of responses and stimuli possible in CAI, it is possible to provide instruction and perform research in the same program.
2. CAI can provide top-flight instruction to large numbers of students. Clearly, a good program can be reproduced and disseminated providing quality instruction to many students.
3. The computer can perform so many functions allied to instruction (record keeping, etc.) that the school can be operated more efficiently and the teacher can be relieved of mundane chores and reserved for his real function: that of instruction.

Unfortunately, at the present time it is almost the case that CAI's potential is its justification. However, with such a fine and worthwhile potential, granting some years of significant developmental work in curricula, and advances in technology, CAI will have a large and important impact on education.

REFERENCES

1. Cooley, W. W., & Glaser, R. The computer and individualized instruction. *Science*, 1969, *166*, 574-582.

2. Hickey, A. E. *Computer-assisted instruction: A survey of the literature* (3rd ed.). Technical Report, No. 8. Newburyport, Massachusetts: ENTELEK, Incorporated, 1968.

3. Atkinson, R. C., & Wilson, H. A. Computer-assisted instruction. In R. C. Atkinson and H. A. Wilson (Eds.), *Computer-assisted instruction: A book of readings*. New York: Academic Press, 1969. Pp. 3-14.

4. Suppes, P. Computer technology and the future of education. In R. C. Atkinson and H. A. Wilson (Eds.), *Computer-assisted instruction: A book of readings*. New York: Academic Press, 1969. Pp. 41-47.

5. Suppes, P., Jerman, M., & Brian, D. *Computer-assisted instruction: Stanford's 1965-66 arithmetic program*. New York: Academic Press, 1968.

6. Atkinson, R. C. Computerized instruction and the learning process. *American Psychologist*, 1968, *23*, 225-239.

7. Suppes, P., & Morningstar, M. Four programs in computer-assisted instruction. In W. Holtzman (Ed.), *Computer-assisted instruction, testing, and guidance*. New York: Harper and Row, in press.

8. Wing, R. *The production and evaluation of three computer-based economics games for the sixth grade*. Final report. Westchester County, New York: Board of Cooperative Educational Services, 1967.

9. Lagowski, J. Computer-assisted instruction in chemistry. In W. Holtzman (Ed.), *Computer-assisted instruction, testing, and guidance*. New York: Harper and Row, in press.

10. Feurzeig, W., Munter, P., Swets, J., & Breen, M. Computer-aided teaching in medical diagnosis. *Journal of Medical Education*, 1964, *39*, 746-754.

11. Feurzeig, W., Papert, S., Bloom, M., Grant, G., & Solomon, C. Programming languages as a conceptual framework for teaching mathematics: Final report on the first fifteen months of the LOGO project. Report No. 1889. Cambridge: Bolt, Beranek, and Newman, 1969.

12. Atkinson, R. C., & Wilson, H. A. (Eds.) Computer-assisted instruction: A book of readings. New York: Academic Press, 1969.

13. Glaser, R. Psychological questions in the development of computer-assisted instruction. In W. Holtzman (Ed.), Computer-assisted instruction, testing, and guidance. New York: Harper and Row, in press.

14. Greeno, J. G. Paired-associate learning with massed and distributed repetition of items. Journal of Experimental Psychology, 1964, 67, 286-295.

15. Fishman, E. J., Keller, L., & Atkinson, R. C. Massed versus distributed practice in computerized spelling drills. In R. C. Atkinson and H. A. Wilson (Eds.), Computer-assisted instruction: A book of readings. New York: Academic Press, 1969. Pp. 227-237.

16. Groen, G. J., & Atkinson, R. C. Models for optimizing the learning process. Psychological Bulletin, 1966, 66, 309-320.

17. Zinn, K. L. Programming conversational use of computers for instruction. Proceedings of the 1968 ACM National Conference. Princeton: Brandon/Systems Press, 1968.

18. Kopstein, F. F., & Seidel, R. J. Computer-administered instruction versus traditionally administered instruction: Economics. In R. C. Atkinson and H. A. Wilson (Eds.), Computer-assisted instruction: A book of readings. New York: Academic Press, 1969. Pp. 327-362.

19. Suppes, P. The application of mathematical models of learning in education. In H. O. A. Wald (Scientific Organizer), Model building in the human sciences. Monaco: Union Európeene D'Editions, 1964. Pp. 39-49.

20. Restle, F. Speed of adding and comparing numbers. Journal of Experimental Psychology, 1970, 83, 274-278.

21. Karush, W., & Dear, R. Optimal stimulus presentation strategy for a stimulus sampling model of learning. Journal of Mathematical Psychology, 1966, 3, 19-47.

22. Suppes, P. Some theoretical models for mathematical learning. Journal of Research and Development in Education, 1967, 7, 5-22.